

复杂对象嵌套结构的规范化^{*}

李天柱

(河北大学计算中心 保定 071002)

摘要 本文将关系模型中的函数依赖、多值依赖、规范化等概念扩展到具有嵌套结构的复杂对象模型,并在函数依赖和多值依赖中融进了实体联系的语义,给出了规范化的方法和步骤.在最后的范式中,不存在不完全嵌套,在嵌套结构的每一层上,仅存在单属性的关键字,相当于关系模型中的BCNF范式.这样的范式可消除冗余,避免更新异常,提高查询效率,保证嵌套结构可简洁而正确地表达实体联系,且不存在平面式的联系类.

关键词 O-O数据库,嵌套结构,函数依赖,规范化.

中图法分类号 TP311

在关系模型中,对同一应用,可生成不同的、但等价的关系模式.大家熟知,通过模式分解可得到不同级别的范式,以求减少和避免更新异常和冗余.

在复杂对象模型中,典型的是在O-O模型中,对象具有更复杂的结构.如果将Oid看作特殊的属性,在平面式的特殊情况,于数据结构上(暂不管方法),对象就近似等效于关系中的元组,类的模式近似于关系模式.类的结构模式不合适,同样会产生更新异常和数据冗余,需要进一步规范化.对象的嵌套的复杂结构表达了实体联系^[1],类结构的规范化应有其自身特点,不仅要消除更新异常和冗余,并能正确而简洁地表达实体联系.

文献[2,3]对NF²关系模型中的模式规范化作了详尽讨论,但NF²关系模型没有对象标识Oid,且那些讨论是基于平面关系的FD和MVD,不完全适用于O-O模型.但一些基本思想可以借鉴.文献[4]讨论了E-R模型的规范化,但只涉及函数依赖,未涉及多值依赖,且是基于关系结构讨论的.文献[5]对O-O模型的模式设计中的规范化做了初步讨论.该文献引入的函数依赖和规范化形式太简单.将规范形式定义为所有函数依赖的所有决定因素全是Oid,将问题过于简化了.在下面的例1(a)中,Oid和学号都是决定因素,但以后将看到,例1(a)是规范的;在例1(c)中,只有Oid是决定因素,但认为它是规范的是不妥的.而且,文献[5]中的规范化方法也过于简单粗糙.

本文将对复杂对象的嵌套结构规范化做更详尽的讨论.为节省篇幅,未全部形式化.

例1:一些O-O模式的例子(为简便和直观,引用属性的值没有使用Oid编码,而代之文字C₁,C₂,C₃).

Oid	学号	姓名	性别	{课程}	Oid	课号	课名	学分
001	9501	张	男	{C ₁ ,C ₂ ,C ₃ }	201	NC ₁	DB	4
002	9502	李	女	{C ₂ ,C ₃ }	202	NC ₂	PG	4
					203	NC ₃	CAI	3

(a) (b)

Oid	学号	姓名	性别	{课程}	Oid	学号	姓名	性别	课号	课名	学分
005	9501	张	男	{C ₁ ,C ₂ }	011	9501	张	男	NC ₁	DB	4
006	9501	张	男	{C ₂ ,C ₃ }	012	9501	张	男	NC ₂	DB	4
007	9502	李	女	{C ₂ ,C ₃ }	013	9501	张	男	NC ₃	DB	3
					014	9502	李	女	NC ₂	DB	4
					015	9502	李	女	NC ₃	DB	3

(c) (d)

图1

图1(a)(b),(c)(d)以不同方式表达了学生选学课程的情况,且所表达的信息相同(两组对象集所含信息相同粗略地指:将每个对象完全平面化后,再做适当连接运算,得到两个相同的元组集合).区别在于:(d)与(a)(b),(c)

* 本文研究得到河北省自然科学基金资助.作者李天柱,1940年生,教授,主要研究领域为数据库,知识库,信息系统.

本文通讯联系人:李天柱,保定 071002,河北大学计算中心

本文 1996-11-12 收到原稿,1997-05-26 收到修改稿

(b)的嵌套结构不同;(a)(b),(c)(b)的模式嵌套结构相同,但对象的构成不同,即对象集不同。也就是说,规范化不仅与类的模式结构有关,还与对象集有关(后面将看到这是由函数依赖决定的)。

图 1(d)实质上是一个 1NF 的平面关系,对学生和课程的更新都会出现更新异常。图 1(c)(b)中,具有嵌套属性,不是平面关系,但对学生张自身信息的更新要涉及两个对象,学生张不再选学课程 C₂ 的更新也涉及两个对象,这些更新异常是由于在(c)中学生信息和 C₂ 有冗余。

图 1(c)不仅可引起更新异常,也会影响查询效率。O-O 模型中的查询的一个重要特点是导航,可加速查询。在图 1(a)(b)中,给定 Oid 为 001,可立即查询到学生张的有关信息,并经一次导航,可查到所学课程情况。但对图 1(c)(b),查到学生张的基本情况和所学课程的有关情况,需对 Oid 为 001 和 002 的两个对象查询,并经二次导航方可完成,这不仅增加了操作的复杂性,也降低了查询效率。产生这些问题的原因也是由于(c)中学生信息有冗余。

此外,图 1(a)(b)能正确而简洁地表达出学生到课程的 1-n 联系语义(事实上表达了学生-课程间 m-n 联系^[1]),但图 1(c)(b)却不能。因此,嵌套结构的规范化程度也影响到嵌套结构对实体联系的表达能力。

本文假设类的属性类型包括原子型(数、字符串等)、元组实体型、实体型集合、非实体型集合。类分为实体型类和非实体型类。前者用以描述客观世界存在的实体,后者用以描述实体间联系,或者依附于某个类而存在的类。非实体型类也叫内嵌联系类。^[1]

1 函数依赖及范式

为了方便起见,称具有原子型域(数、字符串等)的属性为零阶属性,称其他属性(元组型域、集合型域)为高阶属性。本文将 Oid 看作特殊的零阶属性。

1.1 函数依赖

定义 1. 设 X, Y 是类 C 的两个属性(或属性集),且 X 为零阶属性。对类 C 的任两个对象 O₁ 和 O₂,若 O₁. X = O₂. X, 则 O₁. Y = O₂. Y, 称 Y 函数依赖于 X, 记作 X → Y。若 Y 是零阶属性,称此依赖为纯零阶函数依赖,记作 X ^o→ Y; 否则,称此依赖为非纯零阶函数依赖,记作 X ^s→ Y。

纯零阶函数依赖实际上等效于平面关系模型中的 FD。

一个类中,所有属性都函数依赖于 Oid。

在图 1(a)中,学号 ^o→ 性别,学号 ^s→ {课程},后者说明学生对课程的 1-n 联系;但在图 1(c)中,第 2 个函数依赖不成立。

定义 2. 设 X, Y 是类 C 的两个属性集, X 为零阶属性集。若对 X 的任一确定的值,存在一组确定的对象的一组确定的 Y 值与之对应,即至少存在 O₁, O₂ 属于 C, O₁. X = O₂. X, 但 O₁. Y ≠ O₂. Y, 则说 Y 多值依赖于 X, 记作 X → Y。若 Y 为零阶属性,称此为纯零阶多值依赖,记作 X ^o→ Y; 否则,称此为纯零阶多值依赖,记作 X ^s→ Y。

在图 1(c)中,学号 ^s→ {课程};在图 1(d)中,学号 ^o→ {课号, 课名}。

X → Y 表明 X 和 Y 间的 1-n 对应关系,我们更感兴趣的是 X 和 Y 代表实体的情况。

形式上看,属性间或者存在函数依赖或者存在多值依赖,二者必居其一。但函数依赖和多值依赖是一种语义概念,是由用户设定的,不是由给定对象集中各对象的属性取值导出的。在人事管理中,很容易构造一个对象集,使得存在 O₁ 和 O₂, O₁. 性别 = O₂. 性别, 但 O₁. 年龄 ≠ O₂. 年龄, 我们不能由此断言性别 → 年龄, 因为, 当只关心一个人时, 一个人的性别不对应一个年龄的集合, 故用户不会给出性别 → 年龄这个依赖关系。

这里的多值依赖是平面关系模型中多值依赖的扩展。当 X, Y, Z 全是零阶属性集, X → Y, X → Z 同时成立, 且 Z 和 Y 无关, 则在 (X, Y, Z) 范围内, 存在平面关系中的多值依赖 X → Y, X → Z。后面的例 5(b)就表达了平面关系中的多值依赖。

在定义 1 和定义 2 中, 我们使用了 O₁, Y = O₂. Y, 即使用了同一运算 (Identical)。当然, 也可使用浅相等和深相等。本文不讨论后者。

定义 3. 对类 C 中属性集 X, 若对类 C 中的任何属性或属性集 Y, 都有 X → Y 或 X → Y, 则说 X 是类 C 中的准关键字; 若只有 X → Y, 则 X 叫类 C 的关键字。

Oid 永远是关键字。关键字也是准关键字。在一个类中, 作为决定因素, 关键字与 Oid 等效。

定义 4. 在类 C 中, 若存在多值依赖 X → Y, 且 Y 是高阶属性(集合属性), 则称 X → Y 为不完全嵌套依赖, 记作

$X \xrightarrow{U} Y$.

不完全嵌套依赖是一种病态多值依赖,对更新、查询、冗余性以及嵌套化、反嵌套化都有极坏影响.当不完全嵌套依赖存在时,嵌套化和反嵌套化是不可互逆的.

前面所述的函数依赖和多值依赖及关键字诸概念是平面关系模型中相应概念的扩展,因此,也会有部分依赖和传递依赖.

但是,这里的函数依赖和多值依赖又不完全等同于平面关系中的相应概念.若 $X \rightarrow Y$ 中的 Y 是高阶属性(集合属性),则 Y 不能再出现在函数依赖或多值依赖的左部.因此,不会由 Y 过渡而产生传递依赖;当 Y 为零阶属性时,可能引起传递依赖的情况为

$$\begin{aligned} X \rightarrow Y, Y \rightarrow Z &\Rightarrow X \rightarrow Z; & X \rightarrow Y, Y \rightarrow Z &\Rightarrow X \rightarrow Z; \\ X \rightarrow Y, Y \rightarrow Z &\Rightarrow X \rightarrow Z; & X \rightarrow Y, Y \rightarrow Z &\Rightarrow X \rightarrow Z. \end{aligned}$$

另外,函数依赖和多值依赖有如下性质

$$\begin{aligned} (A, B) \rightarrow C, & \text{则 } A \rightarrow C, B \rightarrow C; & (9) \\ (A, B) \twoheadrightarrow C, & \text{则 } A \twoheadrightarrow C, B \twoheadrightarrow C; & (10) \end{aligned}$$

其中 (A, B) 是依赖的最小左部.这与下面的命题不同.

我们不加证明地给出下面命题,供规范化时使用.

若 $(A, B) \rightarrow C$ (或 $(A, B) \twoheadrightarrow C$), 则对确定的 A , 有 $B \rightarrow C$ (或 $B \twoheadrightarrow C$); 对确定的 B , 有 $A \rightarrow C$ ($A \twoheadrightarrow C$). 我们把这样的 $B \rightarrow C$ (或 $B \twoheadrightarrow C$) 叫作 $(A, B) \rightarrow C$ (或 $(A, B) \twoheadrightarrow C$) 的基于 A 的条件依赖, $A \rightarrow C$ ($A \twoheadrightarrow C$) 叫作基于 B 的条件依赖.

另一方面,本文引入的函数依赖和多值依赖的语义描述能力也扩展了平面关系模型中相应概念的描述能力.它们不仅可描述实体属性之间的依赖关系,而且可描述实体联系.这也是我们在复杂对象模型中引入函数依赖及多值依赖的出发点之一.设 X, Y 是两个代表实体(非实体集合)的属性, $X \rightarrow Y$ 且 $Y \rightarrow X$ 描述 X 和 Y 间的 $n-1$ 联系; $X \rightarrow Y$ 且 $Y \rightarrow X$ 描述 X 和 Y 间的 $1-1$ 联系. $X \twoheadrightarrow Y$ 且 $Y \twoheadrightarrow X$ 描述了 X 和 Y 间的 $m-n$ 联系; $X \twoheadrightarrow Y$ 且 $Y \twoheadrightarrow X$ 描述 X 和 Y 间的 $1-n$ 联系. 若 $X \rightarrow Y$, 且 Y 为实体集合型属性, 当对任两个对象 O_1 和 $O_2, O_1, X \neq O_2, X$ 时, $O_1, Y \cap O_2, Y = \emptyset$, 那么, $X \rightarrow Y$ 表示 $1-n$ 实体联系; 当存在两个对象 O_1 和 $O_2, O_1, X \neq O_2, X$ 时, $O_1, Y \cap O_2, Y \neq \emptyset$, 那么, $X \rightarrow Y$ 表示 $m-n$ 联系. 平面关系模型中, 函数依赖和多值依赖均不能如此直接而简洁地描述实体联系. 但在教科书和文献中, 多值依赖和多属性关键字的例子几乎都和实体联系有关.

下面给出一些例子, 因为 Oid 永远是关键字, 故不画出由 Oid 决定的函数依赖, 甚至不标出 Oid . 这些例子较常见, 本文不再做详细说明. 为简单清晰起见, 不使用中文属性名.

例 2: 系-科研项目-研究成员问题(如图 2 所示).

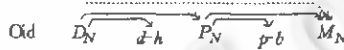


图 2

其中 $D_N, d-h, P_N, p-b, M_N$ 分别为系编号、系主任、研究项目编号、项目预算、项目组成员编号. 此问题中包含部分依赖 $D_N \rightarrow d-h, P_N \rightarrow p-b$, 传递依赖 $D_N \rightarrow M_N$, 而对应于一个确定的 P_N 的 M_N (若干个) 与 D_N 无关. 此例表达了两个 $1-n$ 联系. D_N 为准关键字, (D_N, P_N, M_N) 为关键字.

例 3: 学生-课程-成绩问题——带联系属性的 $m-n$ 联系问题(如图 3 所示).

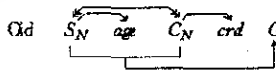


图 3

其中 S_N, age, C_N, crd, G 分别为学号、年龄、课号、课程学分、学生该课程成绩. 图中省略了传递依赖. (S_N, C_N) 为关键字, S_N, C_N 均为准关键字.

例 4: 供应商-工程项目-产品问题——多实体联系(如图 4 所示).



图 4

其中 S, J, P, q 分别为供应商、工程项目、产品和所供产品数量. 注意此例与例 2 不同之处, 即对应于 J 的 P 与 S 有关.

而不是无关,其中 S 和 J 均为准关键字, (S, J, P) 是关键字. 此问题典型地表达了多实体联系,即若干供应商向若干个工程项目供应若干种产品,且不同供应商可向不同工程项目供应不同数量的不同的产品,同一项目可由不同供应商供应产品.

例 5. 供应商-工程项目-产品问题二—— $m-n$ 联系和 $1-n$ 联系(如图 5 所示).

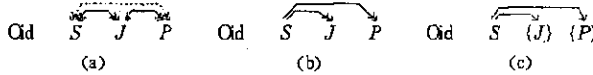


图 5

图 5 中(a)表达两个 $m-n$ 联系,即若干供应商向若干工程项目供应产品,每个工程项目需要若干产品,与供应商无关,且不同项目所需产品可有重的,图 5 中(b)表达了两个 $1-n$ 联系,即一个供应商向若干工程项目供应产品,每个供应商供应若干种产品,图 5 中(c)与(b)表达语义相同,但(c)使用了集合型属性,多值依赖变成了函数依赖. 图 5(a)(b)中的关键字是 (S, J, P) , 图 5(c)中关键字是 S . 图 5(b)所表达的就是平面关系中的 MVD.

例 6. 学生-课程问题之二——不完全嵌套依赖与模式结构和函数依赖有关(如图 6 所示).

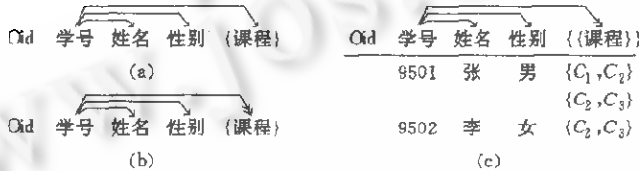


图 6

其中(a)对应于例 1(a), (b)对应于例 1(c), (c)为新构成的. (a), (b)具有相同的模式结构,而(b)含不完全嵌套依赖学号 \rightarrow {课程}, 但(a)不含. 对于(a),若没有严格的约束条件自动检验,不适当的对象生成也可能破坏学号 \rightarrow {课程}, 而导致(b)的情况发生. (b)会导致冗余、更新异常和降低查询效率,这在例 1 中已做了说明. (c)中学号 \rightarrow {{课程}} 是另一种不完全嵌套依赖,其中 C_i 有冗余,且会对更新和查询造成麻烦. 本文不考虑这种不完全嵌套依赖(定义中也不含此情况),因为可以规定不许使用集合的集合型属性. 但(b)中所示的不完全嵌套依赖不得不考虑,因为不适当的对象生成可产生这种不完全嵌套依赖.

1.2 范 式

定义 5. 含有对象标识 Oid 的类 C 是 1NF 的.

前面的例子中的模式均是 1NF 的.

定义 6. 类 C 是 1NF 的,且嵌套结构的顶层不含任何部分依赖、传递依赖、多值依赖、不完全嵌套依赖,则该类 C 是 2NF 的.

后面将看到 2NF 的例子.

定义 7. 若类 C 的任何层都是 2NF 的,则该类 C 是 3NF 的.

例 1(a)(b)中含依赖 $(Oid | 学号) \rightarrow (姓名, 性别, \{课程\})$, $(Oid | 课号) \rightarrow (课名, 学分)$, 此例是 3NF 的(“|”代表“或”的意思).

2 规范化

规范化就是使类模式变成规范形式,规范化过程就是消除部分依赖、传递依赖、多值依赖的过程. 本文中,规范化是通过嵌套化来实现的,而非通过模式分解来实现的.

一个嵌套结构只能直接表达两个实体间的单方向的联系,另一个方向的联系需用另一个类中的另一嵌套结构来表达.^[1]可以依用户欲表达的语义,首先给出一个方向的依赖来表达一个方向的实体联系. 是否和怎样以另一方向的依赖表达另一方向的实体联系,因所采用的是单向方式还是双向方式而不同.^[1]

具有多属性的关键字多数描述实体联系的情况,关键字中每个属性代表一个实体,但一般并不是对象标识(对关键字中多属性不是代表实体,而只是一般属性的情况,可用文献[6]中用户定义的实体标识或 Oid 来做关键字,代替这个多属性关系字). 在多属性关键字中的各属性间不会存在 $X \rightarrow Y$ 式依赖,否则,可缩小关键字;而一般存在多值依赖,如例 3.4 所示. 根据如公式(1)(2)所示的依赖的两个性质,在多属性关键字中可找出一个单属性准关键字(证明略):

下面讨论规范化的方法和步骤。

步骤 1. 消除部分依赖(非主属性部分依赖)。

若只有单属性关键字,则不存在部分依赖,跳过步骤 1;若有多属性关键字,则找出其中一个准关键字。

将其他属性所决定的部分依赖抽出,另组成一个类,其原来位置代之以对该类的引用属性(实体型)。如前面例 3,关键字是 (S_N, C_N) ,取 S_N 为表示实体的准关键字, $S_N \rightarrow age$ 为部分依赖,但要保留 S_N, age ,而摘除 C_N 决定的部分依赖 $C_N \rightarrow crd$,构成一个类叫作 C。这种选择就是将学生看作实体,将课程看作学生的属性。此时,表示 m-n 联系的双值依赖只需取单向的。原模式变成如图 7 所示的形式

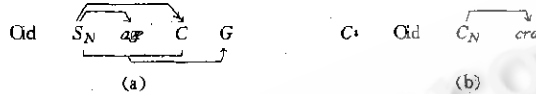


图 7

步骤 2. 消除传递依赖。

将传递从被找出的准关键字(表示实体的)出发的依赖的传递属性及其决定的属性抽出构成一个类,并将这些属性原在位置改成对该类中对象的引用属性。如:图 2(P_N 为传递属性)会变成图 8。



图 8

在图 2 中, (D_N, P_N, M_N) 为关键字, $P_N \rightarrow p-b, P_N \rightarrow M_N$ 既会产生传递依赖,又是部分依赖,按步骤 1 和步骤 2 产生的结果是相同的。

部分依赖和传递依赖的存在,都是由于多个实体及其相关属性存在于同一模式的缘故。

步骤 3. 消除多值依赖。

①若有多值依赖,且此多值依赖所涉及的属性不同时为另一非冗余依赖的左端(对应于既无联系属性,又非多联系的实体联系及其它简单情况),则将多值端嵌套化,并将多值依赖改为函数依赖。如果对对象集已存在,则在做嵌套化的过程中,被嵌套属性下的值可能需做集合并运算,消除重复元素。如此,图 8(a)和图 5(b)分别变为图 9(a)和(b)。



图 9

此步既可消除多值依赖,又可消除不完全嵌套依赖。不过,消除不完全嵌套依赖时模式结构不变,只将多值依赖改为函数依赖,并按上述方法处理对象集及相关属性值。如此,图 6(b)会变成图 6(a),即图 1(c)会变成图 1(a)。

②若有多值依赖,且所涉及属性同时为另一非冗余依赖的左端(对应具有联系属性的实体联系,或多实体联系,关键字是多属性的)。此多值依赖一般是关键字内属性间的依赖。此时,有两种解决方法。

方法 1. 将剩余的部分依赖(准关键字决定的非主属性部分依赖)按步骤 1 的方法抽出。图 7(a)会变成图 10(a)(b)。所得结果中,图 10(a)正是平面关系中的联系关系。文献[7]及 CDMG-93 允许这种形式的模式存在。事实上,按本文定义,这种模式是 1NF 的,仍有缺陷。一是属性值有冗余;二是查询一个学生及所选课程的信息需要作图 10(a)(b)和图 7(b)的连接运算,效率比较低。我们不推荐这种办法,而推荐方法 2。



图 10

方法 2. 将多值依赖的多值端及联系属性(若存在)另立一类,并在它们的原位设置一个对该类的引用属性。另立

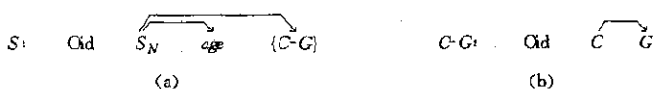


图 11

的新类中会产生条件依赖。如此,图 7(a)变成图 11,图 4 变成图 12。



图12

此时,查询一个学生及所选课程的信息,只要由图 11(a)导航到图 11(b),再导航到图 7(b)即可,查询效率比较高。在图 11(b)和图 12(b)中的依赖是分别由图 7(a)和图 4 生成的条件依赖。图 11(b)中依赖是对图 11(a)中确定的 S_N 而言,图 12(b)中依赖是对图 12(a)中确定的 S 而言。因此,类 $C-G$ 和 $J-P-q$ 均不是独立的实体类,而分别依赖于生成它的类。因此,不宜独立地对 $C-G$ 类或 $J-P-q$ 类进行查询和更新,而要通过生成它的类导航到该类进行上述操作。将图 11(a)中类记作 ST ,则 ST 中对象对 $C-G$ 类中对象的引用可以是共享式的($C-G$ 中没有浅相等的对象),也可以是独立式的($C-G$ 中可能有浅相等的对象)。^[8]这里应该取后者,因为 $C-G$ 中的 $C \rightarrow G$ 对于 ST 中确定的对象成立,对于 ST 中不同的对象, $C \rightarrow G$ 可能不成立。

条件依赖也是依赖,只是成立是有条件的。本文关于依赖的所有讨论对条件依赖仍成立,仍可对包含条件依赖的模式进行规范化。

前述步骤迄今只施于类的顶层。步骤 3 中所得图 11(a)(b)与图 7(b)一起构成 3NF;图 12(a)(b)构成 2NF,因(a)的内嵌类(由图 12(b)所示)还是 1NF 的(有多值依赖和多属性关键字);图 9(a)与图 8(b)所构成的类亦是 2NF 的,因为图 8(b)是内嵌类且是 1NF 的。递归地对新生成的内嵌类施行步骤 1~3。直至各层被嵌套类均是 2NF 的,则就可得到 3NF 的规范化形式。如,继续对图 8(b)施行步骤 3 中的①,得到图 13(a),则图 9(a)与图 13(a)一起构成的模式是 3NF 的;继续对图 12(b)施行步骤 3 中的②,而得到图 13(b)(c),则图 12(a)与图 13(b)(c)构成的类模式也是 3NF 的。

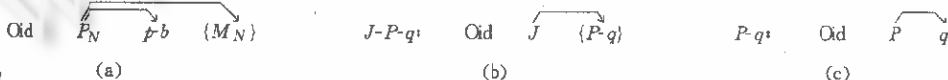


图13

粗略地说,施行步骤 1,2,3,但不施行步骤 3 中②的方法 2,得到的结果相当于平面关系的 3NF 或 4NF;以步骤 3 中②的方法 2 代方法 1,则得到的结果是 3NF 的(按本文定义),且其每一层相当于平面关系的具有单属性关键字的 3CNF 范式。这种范式可保证最小冗余,能正确而简洁地反应实体联系,且便于利用导航提高查询效率。步骤 3 中②的方法 2 实际上相当于对平面关系中的联系类进一步规范化(嵌套化)。不过,在规范化过程中,由所选定的准关键字所决定的部分依赖一直保留着,所得到的最后结果相当于将一个类与相关的联系类合并在一起,并嵌套化。如图 11(a)相当于含有学生类和学生-课程联系类。

3 讨论

复杂对象模型中,嵌套结构的规范化对复杂对象模式的逻辑设计是很重要的。目前对这个问题的研究还较少。文献[9]指出需研究基于 E-R 方法的复杂对象数据库的逻辑设计方法,也就是说复杂对象模式与 E-R 方法亦有密切关系。本文将平面关系模型中的函数依赖和多值依赖、范式、规范化的概念和方法扩展到复杂对象模型中,且建立了函数依赖和多值依赖与实体联系的联系,又考虑到范式中嵌套结构对实体联系正确而简洁的表达和对查询效率的影响,优于文献[5]中的结果。同时指出,仅将 Oid 作为模式中唯一的决定因素是不够的,它不能有效消除冗余和由此引起的更新异常;考虑用户定义的属性构成的决定因素的存在(用户定义的实体标识属性就是一种决定因素),对嵌套结构规范化是有益的。

层次也需规范化,其目的是保证正确表达 isa 语义,确保最大限度的信息继承,这在文献[10]中已做了较详尽的讨论。本文对方法对嵌套结构规范化的影响未加讨论,有待进一步研究。

参考文献

- 1 李天柱. O-O 数据模型中实体联系的表达方法. 计算机研究与发展, 1997, 34(4): 275~280
(Li Tian-zhu. The approaches to express entity relationship in object-oriented model. Computer Research and Development, 1997, 34(4): 275~280)
- 2 Ozsoyoglu Z M et al. A new normal form for nested relations. ACM Transactions on Database Systems, 1987, 12(1): 111~136
- 3 Roth M A, Korth H F. The design of NF² relational databases into nested normal form. ACM SIGMOD, 1987. 143~159
- 4 Rauh O, Stickel E. Standard transformations for the normalization of ER schema. Information Systems, 1996, 21(2): 187~208

- 5 Byung S L. Normalization in OODB design. ACM SIGMOD Record, 1995, 24(3): 23~27
- 6 Li Tian-zhu. Object identity in database systems. Journal of Computer Science and Technology, 1995, 10(4): 380~384
- 7 Kemper A *et al.* Object-oriented database management. Prentice-Hall, 1994
- 8 Cattell R G G. Object data management: Object-oriented and extended relational database system. Addison-Wesley Publishing Company, 1991.
- 9 Blakeley J A, Elmasri R. Object database management: database design and oper. systems. In: Jorge Bocca *et al.* eds. Proceedings of the International Conference on Very Large Data Base'94. Santiago, Chile, 1994. 101~120
- 10 Li Tian-zhu. Normalization of class hierarchy in databases. Journal of Computer Science and Technology, 1996, 11(4): 356~364

Normalization of Nested Structure for Complex Objects

LI Tian-zhu

(Computer Center Hebei University Baoding 071002)

Abstract In this paper, the author extends the concepts of functional dependency, multiple valued dependency, normal form and normalization in relational model to complex object model with nested structure. The semantics of entity relationship are captured in functional dependency and multiple valued dependency. The approaches and steps of normalization are proposed. In final normal form, there exists no any incomplete nesting, and on every level of nested structure there exists only the key with single attribute, which corresponds to the BCNF normal form in relational model. It could not only make redundancy minimum, avoid anomalies of update, but also improve query performance, guarantee to correctly and simply express entity relationship with nested structure and without flat entity relationship classes.

Key words OO database, nested structure, functional dependency, normalization.